

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号  
特開2003-330578  
(P2003-330578A)

(43)公開日 平成15年11月21日(2003.11.21)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 1/26

識別記号

F I  
G 0 6 F 1/00

テーマコード\* (参考)

3 3 4 B 5 B 0 1 1  
3 3 0 F

審査請求 未請求 請求項の数 5 O L (全 10 頁)

(21)出願番号 特願2002-141779(P2002-141779)

(22)出願日 平成14年5月16日(2002.5.16)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72)発明者 五月女 誠

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74)代理人 100094330

弁理士 山田 正紀 (外1名)

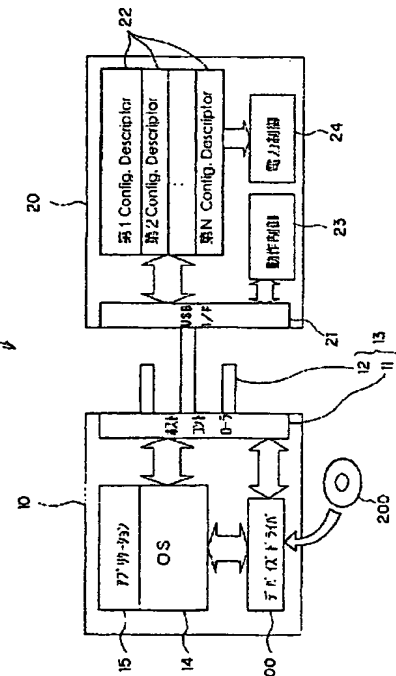
Fターム(参考) 5B011 DB22 DB27 DC01 EA03 EA04  
EB06 FF04 III02 LL14

(54)【発明の名称】 デバイスドライバおよびデバイス

(57)【要約】

【課題】 バスに接続されるデバイス、およびそのデバイスを制御するデバイスドライバに関し、デバイスが強制使用中止(disable)の状態に陥ることを回避することを目的とする。

【解決手段】 デバイスドライバ100は、バス(USB)13に接続されたUSBデバイス20による消費電力の通知に先立って、そのバス13が供給することができる残電力を知得し、知得した残電力が、USBデバイス20によって通知される予定の消費電力を賄うのに十分であるか否かに応じて、USBデバイス20による消費電力の通知を許可してバス13による電力供給を開始させ、あるいは電力不足を回避するための回避処理を実行する。



## 【特許請求の範囲】

【請求項1】 複数デバイスの接続と、接続されたデバイスに対する電力供給とが可能なバスに接続され、該バスに消費電力を通知して電力供給を受ける受給型デバイスを制御するデバイスドライバであって、

前記バスに接続された受給型デバイスによる消費電力の通知に先立って、該バスが供給することができる残電力を知得する残電力知得部と、

前記残電力知得部によって知得された残電力が、前記受給型デバイスによって通知される予定の消費電力を賄うのに十分であるか否かに応じて、該受給型デバイスによる該消費電力の通知を許可して前記バスによる電力供給を開始させ、あるいは電力不足を回避するための回避処理を実行する給電制御部とを備えたことを特徴とするデバイスドライバ。

【請求項2】 前記給電制御部が、前記回避処理として、前記受給型デバイスによる消費電力の通知に先立って所定の表示装置に警告文を表示させ、所定の開始操作を待って、該消費電力の通知を許可して前記バスによる電力供給を開始させるものであることを特徴とする請求項1記載のデバイスドライバ。

【請求項3】 前記給電制御部が、前記回避処理として、前記受給型デバイスから前記バスに前記残電力で賄える消費電力を通知させるとともに、該受給型デバイスをその通知させた消費電力内で動作させるものであることを特徴とする請求項1記載のデバイスドライバ。

【請求項4】 前記受給型デバイスが、前記バスに通知する予定の消費電力を書き換え自在に記憶するものであり、

前記給電制御部は、前記受給型デバイスに記憶されている消費電力を書き換えることにより、その書き換えられた消費電力を該受給型デバイスから前記バスへと通知させるものであることを特徴とする請求項3記載のデバイスドライバ。

【請求項5】 バスに接続され、該バスに消費電力を通知して該バスから電力供給を受けるデバイスであって、前記消費電力が前記バスの供給能力に対して過大である場合に、このデバイスを、該消費電力よりも少ない消費電力で動作する動作状態に移行させる電力制御部と、前記動作状態に移行した場合に、該動作状態における消費電力を前記バスに通知する電力通知部とを備えたことを特徴とするデバイス。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、バスに接続されるデバイスを制御するデバイスドライバ、およびバスから電力の供給を受けるデバイスに関する。

## 【0002】

【従来の技術】 従来より、パーソナルコンピュータなどの本体には、周辺機器（デバイス）を接続するためのバ

スが設けられており、本体内には、そのバスに接続される周辺機器を制御するデバイスドライバが組み込まれている。また、近年では、パーソナルコンピュータのインターフェースとして、Universal Serial Bus（以下USBと称す）が標準的なものとなってきている。

【0003】 このUSBには、複数のデバイス（以下USBデバイスと称す）を接続することができ、そのようなUSBデバイスのうち低電力のUSBデバイスには、USBから電力の供給を受ける方式が採用されたもの（バスパワー・デバイス）も存在する。

【0004】 しかし、USBが普及するに連れて、パーソナルコンピュータに接続されるデバイスのうちUSBデバイスが占める割合が増加してきており、1台のパーソナルコンピュータに接続されるUSBデバイスの数も増加してきている。このため、USBデバイスの合計消費電力が、パーソナルコンピュータに設けられたUSBの許容供給電力を越えてしまう場合がある。USBでは、このような場合にパーソナルコンピュータに異常な負荷が掛かることなどを回避するために、予めUSBデバイスからバス側に消費電力を通知させ、その消費電力がUSBの許容供給電力で賄いきれない場合にはそのUSBデバイスを強制使用中止（disable）の状態にさせるという方式を採用している。

## 【0005】

【発明が解決しようとする課題】 しかし、最近では、携帯電話等といったような、バッテリーを有したデバイスにもUSBデバイスが採用されてきており、そのようなバッテリーを有したUSBデバイスは、電子回路などの動作の電力のみならずバッテリー充電用の電力もUSBから供給を受ける。このような、動作時の消費電力が大きく異なる複数の機能を持ったUSBデバイスは、通常は、最も多く電力を消費する機能における消費電力をバス側に知らせるので、電子回路などの動作だけが必要な場合であっても必要以上の電力をホストに要求してしまい、強制使用中止（disable）の状態に陥ってしまうことがある。

【0006】 また、機能毎に消費電力を切り替えてバス側に知らせるタイプのUSBデバイスも存在するが、消費電力が大きい機能に切り換えた際に強制使用中止（disable）の状態に陥ってしまう場合がある。

【0007】 このような強制使用中止（disable）の状態に一旦陥ってしまうと、USBデバイスの再接続などを行わない限り、その後はそのUSBデバイスを全く利用することができなくなるので、ユーザはUSBデバイスの運用に当たって、強制使用中止（disable）の状態に陥ることがないように慎重に運用することが必要となり、煩雑である。

【0008】 上記の問題は、USBのみならず、複数のデバイスの接続と電力供給とが可能な任意のバスについ

て生じる問題である。

【0009】本発明は、上記事情に鑑み、デバイスが強制使用中止（disable）の状態に陥ることを容易に回避することができるデバイスドライバおよびデバイスを提供することを目的とする。

【0010】

【課題を解決するための手段】上記目的を達成する本発明のデバイスドライバは、複数デバイスの接続と、接続されたデバイスに対する電力供給とが可能なバスに接続され、そのバスに消費電力を通知して電力供給を受ける受給型デバイスを制御するデバイスドライバであって、上記バスに接続された受給型デバイスによる消費電力の通知に先立って、そのバスが供給することができる残電力を知得する残電力知得部と、残電力知得部によって知得された残電力が、受給型デバイスによって通知される予定の消費電力を賄うのに十分であるか否かに応じて、受給型デバイスによる消費電力の通知を許可してバスによる電力供給を開始させ、あるいは電力不足を回避するための回避処理を実行する給電制御部とを備えたことを特徴とする。

【0011】ここで、「通知される予定の消費電力」は、受給型デバイス内に記憶されているものであってもよく、あるいはデバイスドライバ自体に予め組み込まれているものであってもよい。

【0012】本発明のデバイスドライバにおいて、上記給電制御部は、「上記回避処理として、受給型デバイスによる消費電力の通知に先立って所定の表示装置に警告文を表示させ、所定の開始操作を待つて、消費電力の通知を許可してバスによる電力供給を開始させるもの」であってよく、あるいは「上記回避処理として、受給型デバイスからバスに上記残電力で賄える消費電力を通知させるとともに、その受給型デバイスをその通知させた消費電力内で動作させるもの」であってよい。

【0013】警告文を表示して開始操作を待つ回避処理の場合には、受給型デバイスのユーザは、残電力が不足することを警告文によって認識することができ、不要な他のデバイスをバスから外すことなどにより電力を確保してから開始操作を行うことにより強制使用中止（disable）の状態を回避することができる。

【0014】また、残電力で賄える消費電力を通知させる回避処理の場合には、強制使用中止（disable）の状態が自動的に回避されることとなる。

【0015】また、本発明のデバイスドライバは、「上記残電力知得部は、受給型デバイスがバスに接続された際に残電力を知得するものである」という形態であってよく、あるいは「上記受給型デバイスが、互いに消費電力が異なる複数の動作モードを有し、動作モードの変更に伴って新たに消費電力をバスに通知するものであり、上記残電力知得部は、受給型デバイスの動作モードが変更される際に残電力を知得するものである」という

形態であってもよい。

【0016】さらに、本発明のデバイスドライバにおいて、「上記受給型デバイスが、互いに消費電力が異なる複数の動作モードを有するものであり、上記給電制御部は、残電力知得部によって知得された残電力が、上記複数の動作モードそれぞれの消費電力のうち最大の消費電力を賄うのに十分である場合に、受給型デバイスからバスにその最大の消費電力を通知させるものである」ということが好適である。このような形態のデバイスドライバによれば、残電力が十分に大きい時点で受給型デバイス用の電力が予め確保されるので、以後、どのような動作モードが選択されても強制使用中止（disable）の状態は回避される。

【0017】なお、本発明のデバイスドライバは、「上記受給型デバイスが、バスに通知する予定の消費電力を複数個記憶するものであり、上記給電制御部は、受給型デバイスに記憶されている複数の消費電力のうちの1つの消費電力を選択することにより、その選択された消費電力を受給型デバイスからバスへと通知させるものである」という形態であってもよいが、「上記受給型デバイスが、バスに通知する予定の消費電力を書き換え自在に記憶するものであり、上記給電制御部は、受給型デバイスに記憶されている消費電力を書き換えることにより、その書き換えられた消費電力を受給型デバイスからバスへと通知させるものである」という形態が好適である。このような形態のデバイスドライバによれば、受給型デバイスに用意するべきメモリ素子を少なく抑えることができ、受給型デバイスのコストを抑えることができる。上記目的を達成する本発明のデバイスは、バスに接続され、そのバスに消費電力を通知してそのバスから電力供給を受けるデバイスであって、上記消費電力がバスの供給能力に対して過大である場合に、このデバイスを、その消費電力よりも少ない消費電力で動作する動作状態に移行させる電力制御部と、上記動作状態に移行した場合に、その動作状態における消費電力をバスに通知する電力通知部とを備えたことを特徴とする。

【0018】本発明のデバイスによれば、消費電力がバスの供給能力を超えた場合に動作状態が移行して消費電力が抑えられるので、強制使用中止（disable）の状態は回避される。

【0019】

【発明の実施の形態】以下、本発明の実施形態について説明する。

【0020】図1は、本発明のデバイスドライバの一実施形態が適用されるコンピュータシステムを示す図である。

【0021】この図には、コンピュータ本体10とUSBデバイス20とを有するコンピュータシステム1が示されており、このコンピュータシステム1は、図示が省略されたディスプレイなども有している。

【0022】コンピュータ本体10には、ホストコントローラ11と、そのホストコントローラ11に制御される3組のバスライン12とを備えたUSB13が設けられており、バスライン12は、コネクタとケーブルで構成されている。USBデバイス20は、このUSB13のバスライン12の1つにUSB\_I/F21を介して接続されている。このUSBデバイス20は、本発明にいう受給型デバイスの一例であって、USB13から電力の供給を受ける。3組のバスライン12を枝分かれさせることにより、USB13には最大で127個のデバイス10を接続することができ、USB13は、所定の上限電力内であれば、複数の受給型デバイスに電力を供給することもできる。

【0023】コンピュータ本体10には、コンピュータシステム1全体の基本的な管理を行うOS(Operating System)14と、コンピュータシステム1を所定の役目の装置として動作させるアプリケーション15と、USBデバイス20を制御する、そのUSBデバイス20用のデバイスドライバ100がインストールされている。ここで、このデバイスドライバ100は、CD-ROM200に記憶されていたものであり、そのCD-ROM200が、コンピュータシステム1に備えられた、図示が省略されたCDドライブに装填されてコンピュータ本体10に読み込まれることによってインストールされたものである。このデバイスドライバ100は、本発明のデバイスドライバの一実施形態である。

【0024】図2は、デバイスドライバの構成を示す図である。

【0025】この図2には、CD-ROM200に記憶された状態でのデバイスドライバ100の構成が示されているが、コンピュータ本体10にインストールされた状態でのデバイスドライバ100も同様な構成を有しており、デバイスドライバ100には、残電力知得部110と給電制御部120が備えられている。

【0026】残電力知得部110は、図1に示すUSB13が供給することができる電力の残りを知得するものであり、給電制御部120は、USBデバイス20に対する電力供給を制御するものである。

【0027】図1に戻って説明を続ける。

【0028】USBデバイス20にはN種類の動作モードが存在し、コンフィグレーション・デスク립タ22と称されるメモリ素子がN個備えられており、これらのコンフィグレーション・デスク립タ22それぞれには、N種類の動作モードそれぞれにおける動作環境(コンフィグレーション)が記憶されている。コンフィグレーション・デスク립タ22の記憶内容の詳細については後述する。また、USBデバイス20には、上述したN種類の動作モードの切替えや、動作モードに応じたコンフィグレーション・デスク립タ22の選択等を担う

動作制御ブロック23と、選択されたコンフィグレーション・デスク립タ22に記憶されている動作環境に従う電力制御を担う電力制御ブロック24も備えられている。動作制御ブロック23は、本発明のいう電力通知部の一例に相当し、電力制御ブロック24は、本発明のいう電力制御部の一例に相当する。

【0029】図3は、コンフィグレーション・デスク립タ22の記憶内容を示す図である。

【0030】この図3には、コンフィグレーション・デスク립タの記憶内容がコンフィグテーブル300にまとめられて示されており、コンフィグレーション・デスク립タには、0番から8番までの9つのフィールドが記憶されている。また、このコンフィグテーブル300には、それら9つのフィールドそれぞれについて、フィールド名301、サイズ302、フィールドの値303、およびフィールドの記憶内容304が示されている。これら9つのフィールドのうち8番のフィールド310にはMaxPowerというフィールド名が与えられており、該当する動作モードにおける最大の消費電力が記憶されている。

【0031】図1に示すUSB13には、命令(リクエスト)体系が備えられており、デバイスドライバ100は、この命令体系を用いてUSBデバイス20を制御する。

【0032】この命令体系では、USBデバイスの種類にかかわらず共通に規定された13個のスタンダード・リクエストと、USBデバイスをいくつかのクラスに分類した各クラス内で共通に規定された1個のクラス・リクエストと、USBデバイスのメーカーが自由に定義することができる1個のベンダー・リクエストが用いられる。

【0033】図4は、USBに備えられたスタンダード・リクエストを表す図である。

【0034】この図4には、USBに備えられた13個のスタンダード・リクエストがリクエストテーブル320にまとめられて示されており、このリクエストテーブル320には、13個のスタンダード・リクエストそれぞれについて、リクエストナンバー321と、リクエスト名322と、リクエストの内容323が示されている。

【0035】以下の説明では、ここに示す13個のスタンダード・リクエストのうち、6番のリクエスト330と、9番のリクエスト340が用いられる。6番のリクエスト330は、GET\_DESCRIPTORというリクエスト名を有しており、コンフィグレーション・デスク립タに記憶されている内容を取得する命令である。また、9番のリクエスト340は、SET\_CONFIGURATIONというリクエスト名を有しており、コンフィグレーション・デスク립タを指定することによりUSBデバイスの動作環境(コンフィグレーション

ョン)を設定する命令である。

【0036】以下、本発明のデバイスドライバの第1実施形態における制御動作手順について説明する。なお、以下の説明では、図1に示す構成要素を適宜に参照する。

【0037】図5は、第1実施形態における制御動作手順の前段を示す図であり、図6は、第1実施形態における制御動作手順の後段を示す図である。

【0038】デバイスドライバ100によるUSBデバイス20の制御は、コンピュータ本体10の電源がオン状態になったとき、あるいはUSBデバイス20がUSB13に接続(Plug IN)されたときに開始され、先ず、USBデバイス20からホストコントローラ11に、バスライン12を介して所定の信号が通知され(ステップS101)、その信号を端緒としてOS14などによって、USBデバイス20を認識するための前処理(ステップS102)が実行される。

【0039】この前処理が終了すると、デバイスドライバ100が、図4に示すGET\_DESCRIPTORというリクエスト330をUSBデバイス20に投げて(ステップS103)、このUSBデバイス20から、USBデバイス20内に備えられたコンフィグレーション・デスクリプタ22の数と、各コンフィグレーション・デスクリプタ22に記憶されているMaxPower値とを取得する(ステップS104)。このように取得されたMaxPower値は、デバイスドライバ100によってコンピュータ本体10内に保管される。

【0040】このような前段の手順が終了した後、図6に示す後段の手順は、USBデバイス20の認識後に初めて、USBデバイス20が有する複数の動作モードから1つの動作モードが指定されて開始された場合や、1つの動作モードから別の動作モードに切り替えられた場合に開始される。なお、これらの場合を包括して、以下では単に「動作モードの切替え」と称する。

【0041】動作モードの切替えは、以下の2つ要求のうちのいずれかの要求によって発生する。1つは、USBデバイス20からの切替え要求(ステップS201)であり、もう一つはアプリケーション15からの切替え要求(ステップS202)である。

【0042】USBデバイス20からの切替え要求は、USBデバイス20内の動作制御ブロック23から、USB\_I/F21、バスライン12、USBホストコントローラを順に経由して、デバイスドライバ100に通知される。

【0043】また、アプリケーション15からの切替え要求は、コンピュータ本体10内のアプリケーション15から、OS14を通じてデバイスドライバ100に通知される。

【0044】これらの要求を受けて、デバイスドライバ100の残電力知得部110(図2参照)は、ホストコ

ントローラ11からOS14を介して、供給可能な残りの電力値(許容供給電力値)を知得する(ステップS203)。その後、デバイスドライバ100の給電制御部120(図2参照)は、図5に示す手順で事前に取得されて保管されたMaxPower値のうち、切替え後の動作モードに対応するMaxPower値と、残電力知得部110によって知得された許容供給電力値とに基づいて、切替え後の動作モードに必要な消費電力が許容供給電力値で賄えるか否かを判定する。即ち、動作モードの切替に伴うMaxPower値の増加分が許容供給電力値を越えているか否かを判定する。

【0045】ここで、MaxPower値の増加分が許容供給電力値を越えていないと判定された場合(動作モードの切替に伴ってMaxPower値が減少する場合を含む)には、デバイスドライバ100は、図4に示すSET\_CONFIGURATIONというリクエスト340をUSBデバイス20に投げて(ステップS204)、新たな動作モードの動作環境(コンフィグレーション)にUSBデバイス20を移行させる。動作環境が移行したUSBデバイス20は、新たな動作環境での消費電力をホストコントローラ11を介してOS14に通知し、USB13から動作電力の供給を受ける。また、デバイスドライバ100は、OS14を介してアプリケーション15に、動作モードの移行(切替え)が完了したことを通知する(ステップS205)。

【0046】一方、給電制御部120によって、MaxPower値の増加分が許容供給電力値を越えていると判定された場合には、動作モードの移行をする／しないを、デバイスドライバ100がOS14およびアプリケーション15を介してユーザに問い合わせ(ステップS206)、Yes(移行する)／No(移行しない)を表す返答通知を受ける(ステップS207)。このとき、問い合わせを受けたユーザは、USB13に接続されている余分なUSBデバイスを外すことなどによって電力を確保した上でYes(移行する)を表す返答通知を返すか、あるいは、新たな動作モードへの切替えをあきらめてNo(移行しない)を表す返答通知を返す。

【0047】給電制御部120は、No(移行しない)を表す返答通知を受けた場合には、新たな動作モードへの切替えを禁止し、Yes(移行する)を表す返答通知を受けた場合には、SET\_CONFIGURATIONというリクエスト340をUSBデバイス20に投げて(ステップS208)、新たな動作モードの動作環境(コンフィグレーション)にUSBデバイス20を移行させる。そして、上記同様に、USBデバイス20は消費電力の通知を行って電力の供給を受け、デバイスドライバ100は、OS14を介してアプリケーション15に、動作モードの移行(切替え)が完了したことを通知する(ステップS209)。

【0048】このような第1実施形態によれば、デバイ

スが強制使用中止 (disable) の状態に陥る前に、動作モードの移行をする／しないの問い合わせがユーザに提示されるので、ユーザが適宜に対処することにより強制使用中止 (disable) の状態を容易に回避することができる。

【0049】次に、本発明のデバイスドライバの第2実施形態について説明する。

【0050】この第2実施形態は、図1に示すN個のコンフィグレーション・デスク립タ22のうち、例えば第N番目のコンフィグレーション・デスク립タが、Max Power 値が書き換え可能なものであるという点を除いて、図1に示すコンピュータシステム1と同等なコンピュータシステムを前提とする。また、この第2実施形態は、図2に示す構成と同様の構成を有している。

【0051】以下では、図1～図4を、第2実施形態に関する図として流用して説明する。

【0052】図7は、第2実施形態における制御動作手順を示す図である。

【0053】第2実施形態でも、デバイスドライバ100によるUSBデバイス20の制御は、コンピュータ本体10の電源がオン状態になったとき、あるいはUSBデバイス20がUSB13に接続 (Plug IN) されたときに開始され、先ず、USBデバイス20からホストコントローラ11に、バスライン12を介して所定の信号が通知され (ステップS301)、その信号を端緒としてOS14などによって、USBデバイス20を認識するための前処理 (ステップS302) が実行される。

【0054】この前処理が終了すると、第2実施形態では、デバイスドライバ100の残電力知得部110 (図2参照) によって、OS14から許容供給電力値が取得され、その許容供給電力値が、デバイスドライバ100の給電制御部120 (図2参照) によって、OS14、ホストコントローラ11、バスライン12、USB\_I/F21を順次に経由して、第N番目のコンフィグレーション・デスク립タ22へと伝達されて、その第N番目のコンフィグレーション・デスク립タにMax Power 値として設定される (ステップS304)。このようなMax Power 値の設定は、例えば、上述したベンダー・リクエストを定義することによって実現することができる。

【0055】その後、デバイスドライバ100の給電制御部120は、SET\_CONFIGURATIONリクエストを用いて、USBデバイス20の動作環境を、第N番目のコンフィグレーション・デスク립タ22に記憶されている動作環境に切り替える (ステップS305)。

【0056】USBデバイス20の電力制御ブロック24は、切り替えられた動作環境に設定されているMax Power 値を越えることの無い様に電力制御を行う。

例えば、USBデバイス20が、USB13から充電電力を供給される充電電池を備えたものであり、許容供給電力値が通常の充電電力を下回っているといった場合には、電力制御ブロック24は、設定されているMax Power 値 (即ち許容供給電力値) 内でゆっくりと充電を行うといった電力制御を行うこととなる。

【0057】このような第2実施形態によれば、強制使用中止 (disable) の状態が自動的に回避される。

【0058】なお、この第2実施形態では、ステップS304で許容供給電力値に余裕があって、USBデバイス20が消費する最大の消費電力値を上回っている場合には、その最大の消費電力値がMax Power 値として設定される。この最大の消費電力値は、デバイスドライバ100内に予め用意されている値である。

【0059】このように最大の消費電力値がMax Power 値として設定されることにより、過大な電力の割り当てが回避されるとともに、以後、どのような動作モードに切り替わっても強制使用中止 (disable) の状態が回避されることとなる。

【0060】次に、本発明のデバイスドライバの第3実施形態について説明する。

【0061】図8は、第3実施形態が適用されるコンピュータシステムを示す図である。

【0062】この図8に示すコンピュータシステム2は、図1に示すN個のコンフィグレーション・デスク립タ22を有するUSBデバイス20に替えて、Max Power 値が書き換え可能な1つのコンフィグレーション・デスク립タ26を有するUSBデバイス25を備えている点と、コンピュータ本体10に第3実施形態のデバイスドライバ150が組み込まれている点を除いて、図1に示すコンピュータシステム1と同様である。また、第3実施形態のデバイスドライバ150の構成は、図2に示すデバイスドライバ100構成と同様である。

【0063】以下では、図2～図4を、第3実施形態に関する図として流用して説明する。

【0064】図9は、第3実施形態における制御動作手順を示す図である。

【0065】第3実施形態でも、デバイスドライバ150によるUSBデバイス20の制御は、コンピュータ本体10の電源がオン状態になったとき、あるいはUSBデバイス20がUSB13に接続 (Plug IN) されたときに開始され、先ず、USBデバイス20からホストコントローラ11に、バスライン12を介して所定の信号が通知され (ステップS401)、その信号を端緒としてOS14などによって、USBデバイス20を認識するための前処理 (ステップS402) が実行される。

【0066】この前処理が終了すると、第3実施形態で

も第2実施形態と同様に、デバイスドライバ150の残電力知得部110（図2参照）によって、OS14から許容供給電力値が取得され、デバイスドライバ150の給電制御部120（図2参照）によって、OS14、ホストコントローラ11、バスライン12、USB\_I/F21を介して、コンフィグレーション・デスクリプタ26のMax Power値が、第2実施形態のステップS304における設定値と同様の値に設定される（ステップS403）。

【0067】その後、第3実施形態では、デバイスドライバ150から、OS14およびホストコントローラ11を経由してUSBのバス・リセットが発せられて（ステップS404）、上記同様の前処理（ステップS405）が実行され、USBデバイス25が再度認識される。その後、デバイスドライバ150の給電制御部120が、SET\_CONFIGURATIONリクエストを用いて、USBデバイス25の動作環境を、コンフィグレーション・デスクリプタ26に記憶されている動作環境に設定する（ステップS406）。

【0068】このような第3実施形態によれば、第2実施形態における給電制御と同等な給電制御を、1つのコンフィグレーション・デスクリプタ26だけで実現することができ、USBデバイス25のコストを抑えることができる。

【0069】なお、第3実施形態におけるステップS403～ステップS406の手順で、上記第1実施形態におけるステップS204やステップS208を置き換えるといった応用によって、第1実施形態における給電制御と同等な給電制御を1つのコンフィグレーション・デスクリプタ26だけで実現することもできる。

【0070】以下、本発明の種々の形態を付記する。

【0071】（付記1）複数デバイスの接続と、接続されたデバイスに対する電力供給とが可能なバスに接続され、該バスに消費電力を通知して電力供給を受ける受給型デバイスを制御するデバイスドライバであって、前記バスに接続された受給型デバイスによる消費電力の通知に先立って、該バスが供給することができる残電力を知得する残電力知得部と、前記残電力知得部によって知得された残電力が、前記受給型デバイスによって通知される予定の消費電力を賄うのに十分であるか否かに応じて、該受給型デバイスによる該消費電力の通知を許可して前記バスによる電力供給を開始させ、あるいは電力不足を回避するための回避処理を実行する給電制御部とを備えたことを特徴とするデバイスドライバ。

【0072】（付記2）前記給電制御部が、前記回避処理として、前記受給型デバイスによる消費電力の通知に先立って所定の表示装置に警告文を表示させ、所定の開始操作を待って、該消費電力の通知を許可して前記バスによる電力供給を開始させるものであることを特徴とする付記1記載のデバイスドライバ。

【0073】（付記3）前記給電制御部が、前記回避処理として、前記受給型デバイスから前記バスに前記残電力で賄える消費電力を通知させるとともに、該受給型デバイスをその通知させた消費電力内で動作させるものであることを特徴とする付記1記載のデバイスドライバ。

【0074】（付記4）前記残電力知得部は、前記受給型デバイスが前記バスに接続された際に残電力を知得するものであることを特徴とする付記1記載のデバイスドライバ。

【0075】（付記5）前記受給型デバイスが、互いに消費電力が異なる複数の動作モードを有し、動作モードの変更に伴って新たに消費電力を前記バスに通知するものであり、前記残電力知得部は、前記受給型デバイスの動作モードが変更される際に残電力を知得するものであることを特徴とする付記1記載のデバイスドライバ。

【0076】（付記6）前記受給型デバイスが、互いに消費電力が異なる複数の動作モードを有するものであり、前記給電制御部は、前記残電力知得部によって知得された残電力が、前記複数の動作モードそれぞれの消費電力のうち最大の消費電力を賄うのに十分である場合に、前記受給型デバイスから前記バスにその最大の消費電力を通知させるものであることを特徴とする付記1記載のデバイスドライバ。

【0077】（付記7）前記受給型デバイスが、前記バスに通知する予定の消費電力を複数個記憶するものであり、前記給電制御部は、前記受給型デバイスに記憶されている複数の消費電力のうちの1つの消費電力を選択することにより、その選択された消費電力を該受給型デバイスから前記バスへと通知させるものであることを特徴とする付記3または6記載のデバイスドライバ。

【0078】（付記8）前記受給型デバイスが、前記バスに通知する予定の消費電力を書き換え自在に記憶するものであり、前記給電制御部は、前記受給型デバイスに記憶されている消費電力を書き換えることにより、その書き換えられた消費電力を該受給型デバイスから前記バスへと通知させるものであることを特徴とする付記3または6記載のデバイスドライバ。

【0079】（付記9）バスに接続され、該バスに消費電力を通知して該バスから電力供給を受けるデバイスであって、前記消費電力が前記バスの供給能力に対して過大である場合に、このデバイスを、該消費電力よりも少ない消費電力で動作する動作状態に移行させる電力制御部と、前記動作状態に移行した場合に、該動作状態における消費電力を前記バスに通知する電力通知部とを備えたことを特徴とするデバイス。

【0080】

【発明の効果】以上説明したように、本発明のデバイスドライバやデバイスによれば、強制使用中止（disable）の状態が容易に回避され、デバイスが常に動作

13

14

し続けることが可能となる。

【図面の簡単な説明】

【図1】本発明のデバイスドライバの一実施形態が適用されるコンピュータシステムを示す図である。

【図2】デバイスドライバの構成を示す図である。

【図3】コンフィグレーション・デスクリプタ22の記憶内容を示す図である。

【図4】USBに備えられたスタンダード・リクエストを表す図である。

【図5】第1実施形態における制御動作手順の前半を示す図である。

【図6】第1実施形態における制御動作手順の後段を示す図である。

【図7】第2実施形態における制御動作手順を示す図である。

【図8】第3実施形態が適用されるコンピュータシステムを示す図である。

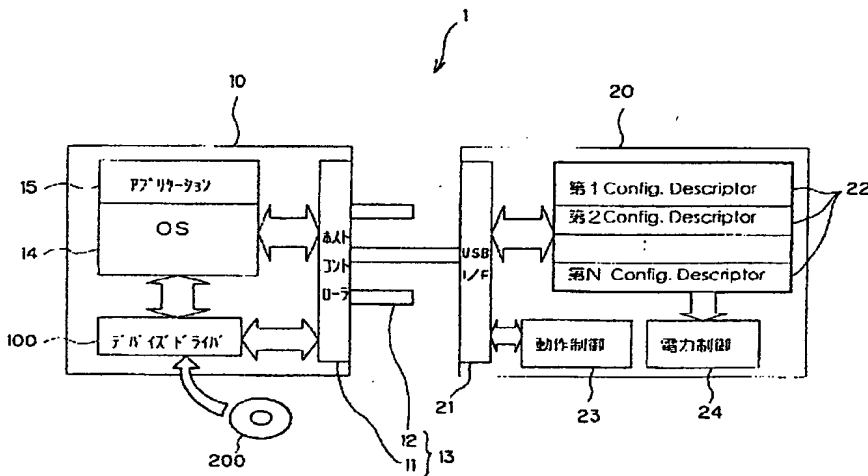
【図9】第3実施形態における制御動作手順を示す図である。

【符号の説明】

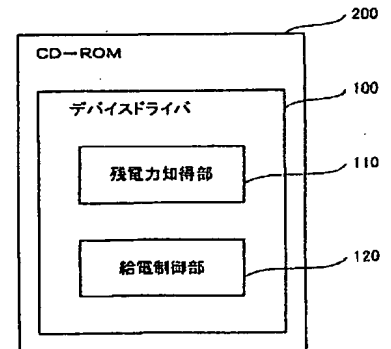
- 1, 2    コンピュータシステム  
10    コンピュータ本体  
11    ホストコントローラ  
12    バスライン  
13    USB  
14    OS  
15    アプリケーション  
20, 25    USBデバイス  
21    USB\_I/F  
22, 26    コンフィグレーション・デスクリプタ  
         (メモリ)  
23    動作制御ブロック  
24    電力制御ブロック  
100, 150    デバイスドライバ  
110    残電力知得部  
120    給電制御部  
200    CD-ROM  
300    コンフィグテーブル  
320    リクエストテーブル

20

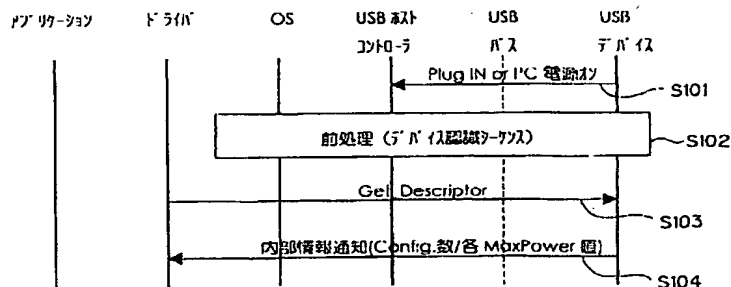
【図1】



【図2】



【図5】





【図3】

300

	Field	サイズ	値	内容
0	bLength	1バイト	Number	このデスクリプタのサイズ
1	bDescriptor type	1バイト	Constant	コフィギュレーション
2	wTotalLength	2バイト	Number	このコフィギュレーションの長さ
4	bNumInterfaces	1バイト	Number	サポートされるインターフェイス数
5	bConfiguration Value	1バイト	Number	このコフィギュレーションを選択する「Set Configuration」への引数として使う値
6	iConfiguration	1バイト	Index	文字列デスクリプタのインデックス
7	bmAttributes	1バイト	Bitmap	コフィギュレーション特性
8	MaxPower	1バイト	mA	最大消費電力 (2mA単位で表記)

310

301      302      303      304

【図4】

320

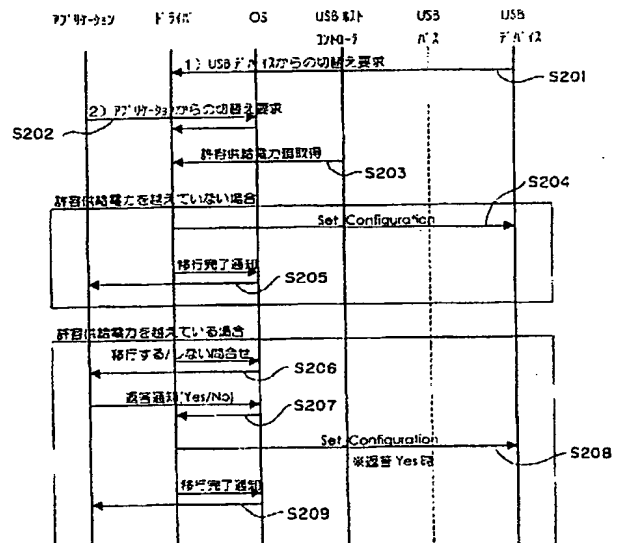
	コマンド・リスト	内容
0	GET_STATUS	ステータスの取得
1	CLEAR_FEATURE	特定機能のクリアまたは、ディセーブル
3	SET_FEATURE	特定機能の設定または、イネーブル
5	SET_ADDRESS	デバイスアドレスの設定
6	GET_DESCRIPTOR	指定デスクリプタの取得
7	SET_DESCRIPTOR	指定デスクリプタの設定
8	GET_CONFIGURATION	現在のデバイスのコフィギュレーション値の取得
9	SET_CONFIGURATION	デバイス構成の設定
10	GET_INTERFACE	インターフェイスの取得
11	SET_INTERFACE	インターフェイスの設定
12	SYNCH_FRAME	エンドポイント同期パルスを送信

330

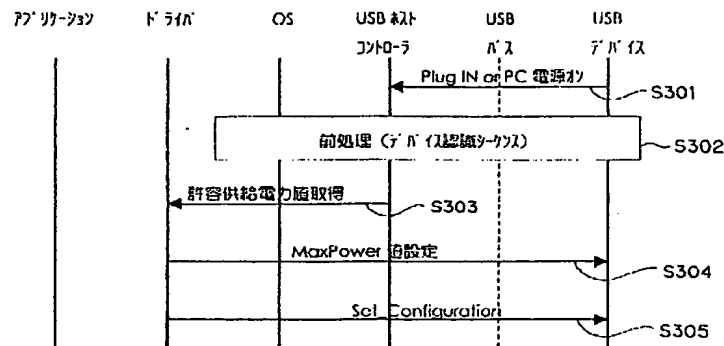
340

321      322      323

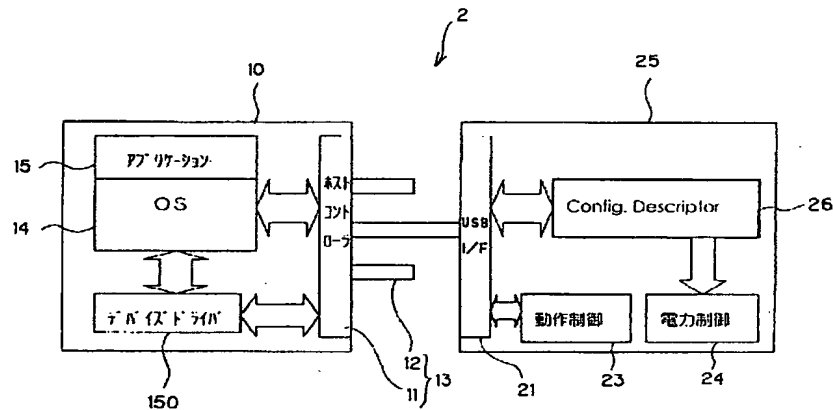
【図6】



【図7】



【図8】



【図9】

